Image Super Resolution

Mubashira km
Roll no: 30
Product Owner: Vasudevan T V

Table of Contents

1. Description
2. Modules
3. Methodology
4. Developing Environment
5.Future Enhancement
6. Product backlog
7. User story
8. Project plan
9. Sprint plan
10. Sprint Actual

Description

- High-resolution (HR) image reconstruction from single low-resolution (LR) image is one of the important vision applications.
- Despite numerous algorithms have been successfully proposed in recent years, efficient and robust single-image super-resolution (SR) reconstruction is still challenging by several factors, such as inherent ambiguous mapping between the HR-LR images, necessary huge exemplar images, and computational load.
- In this paper, we proposed a new learning-based method of single-image SR.
- Inspired by simple mapping functions method, a mapping matrix table of HR-LR feature patches is calculated in the training phase.
- Each atom of dictionary learned from LR feature patches is corresponding to a mapping matrix in the mapping matrix table.
- Combining this mapping table with sparse coding, high quality and HR images are reconstructed in reconstruction phase.
- The effectiveness and efficiency of this method is validated with experiments on the training datasets.
- Compared with state-of-art methods, jagged and blurred artifacts are depressed effectively and high reconstruction quality is acquired with less exemplar images.

MODULES

Module 1

Input Imaging

Preprocessing Image

Module 2

Data collection and preprocessing

Module 3

Comparison and Algorithm implementation

Module 4

Detection and Modifications

- edge detection
- replacing pixel data
- writing image
- show/save

Methodology

- Image super resolution refers to enhancing the resolution of an image from low-resolution to high-resolution
- There are six algorithms used "sr", "esr", "dsr", "ddsr", "rnsr", "distilled_rnsr“.
- Super resolution(sr) refers to methods aiming at increasing spatial resolution of digital images.

Detection and modifications

- Tensorflow is a python-friendly open source library for numerical computation that make machine learning faster and easier
- Python is easy to learn and work with and provides convenient ways to express how high-level abstractions can be coupled together
- Edge detection is a technique of image processing used to identify points in a digital image with discontinuities, simply to say, sharp changes in the image brightness. These points where the image brightness varies sharply are called the edges (or boundaries) of the image.
- It used to upscale low-resolution images to a higher resolution to fit the display of high-resolution monitors. The catch was that the upscaled image showed quality similar to that of rendering the image natively in a higher resolution.

- **Super Resolution** is an umbrella term for a class of techniques in which accurate or close-to-accurate pixel information is added to construct a high-resolution image from its low-resolution form while maintaining its original quality.
- There are two section,main and models
- In main section is a run file
- In models,the algorithms are compared and implemented

Class creation,each class contain various algorithms which have its own method .

Super-resolution refers to the **process of upscaling or improving the details of the image**. ... The image given above illustrates super-resolution. The original high-resolution image shows the best details when zoomed in. The other images are achieved after reconstruction after using various super-resolution methods

- The lower resolution input image to be upscaled
- The input image upscaled by nearest neighbour interpolation
- The input image upscaled by bi-linear interpretation, this is what your Internet browser would typically need
- The input image upscaled and improved by this model’s prediction
- The target image or ground truth, which was downscaled to create the lower resolution input.
- Writing image
- We got a high resolution image.show and save the

Image.

DEVELOPING ENVIRONMENT

Hardware specification:

- i3 Processor Based Computer or higher
- Memory: 1 GB RAM
- Hard Drive: 50 GB
- Monitor
- Internet Connection

Software specification:

- Language :Python
- Front end : Python
- Back end : python
- Operating system : windows 7 or higher
- IDE : PyCharm,Anaconda(spyder)

Future Enhancement

- So what should we do in further studies? More advanced, adaptive, and faster methods with extensive applicability are always desirable. In addition, methods should be closely combined with actual requirements. The rapid development of hardware devices will also bring new challenges to the application of the SR framework. For instance, the Google Skybox project will provide us with an opportunity to obtain real-time HR “earth-observation videos” using remotely-sensed image SR. The concept of SR has also been extended to related fields such as fluorescence microscopy [17,207–209] and multi-baseline tomographic synthetic aperture radar (SAR) imaging [210,211]. Moreover, researchers have attempted to apply the single-frame SR techniques to the processing of medical and remote sensing imagery. However, the practicability of these methods is still limited by the relatively poor performance and time consumption, and acceleration strategies are essential for large-scale applications. In conclusion, the future of SR is still in our hands.

PRODUCT BACKLOG

User Story ID	Priority <High/ Medium / Low	Size(hours)	Sprint	Status/ Planned/ In progress/completed	Release date	Release goal
1	medium	5	1	completed	27/12/2021	Input imaging
2	medium	5		completed	28/12/2021	Preprocessing image
3	medium	5	2	planned	15/01/2022	Data collection and preprocessing
4	high	10	3	planned	23/01/2022	Comparison and algorithm Implementation
5	high	10	4	planned	26/01/2022	Edge detection
6	high	10		planned	05/02/2022	Replacing pixel datas

USER STORY

User story ID	As a<type of user>	I want to <perform some task>	So that I can <achieve some goal>
1	User	Input imaging	Read image from a system
2	User	Preprocessing image	Resize and color scheme changes as per project requirement
3	User	Data collection and preprocessing	Cleaned final dataset
4	User	Comparison and algorithm implementation	Compare different type algorithm and implement
5	User	Edge detection	Finding boundaries Of objects within images
6	User	Replacing pixel	Spreaded pixel data

Project Plan

User	Task name	Start date	End date	Days	Status
1	Sprint 1	27/12/2021	27/12/2021	2 days	completed
2		28/12/2021	28/12/2021		completed
3	Sprint2	15/01/2022	16/01/2022	2 days	planned
4	Sprint 3	23/01/2022	24/01/2022	2 days	planned
5	Sprint4	26/01/2022	27/01/2022	6 days	planned
6		5/02/2022	6/02/2022		planned
7		12/02/2022	13/02/2022		planned

Sprint plan

Backlog items(user story	completion	Estimated hrs	day 1	day2	day3	day4	day5	day6	day7	day8	day9	day10	day11	day12
Input imaging	27/12/2021	5	5		0	0	0	0	0	0	0	0	0	0
Preprocessing image	28/12/2021	5	0	5	0	0	0	0	0	0	0	0	0	0
Data collection and preprocessing	15/01/2022	5	0	0	2	3	0	0	0	0	0	0	0	0
Comparison and algorithm implementation	23/01/2022	10	0	0	0	0	5	5	0	0	0	0	0	0
Edge detection	26/01/2022	10	0	0	0	0	0	0	5	5	0	0	0	0
Replacing pixel data	05/02/2022	10	0	0	0	0	0	0	0	0	5	5	0	0
Writing image Show/save	12/02/2022	5	0	0	0	0	0	0	0	0	0	0	3	2
total		50	5	5	2	3	5	5	5	5	5	5	3	2

Sprint actual

Backlog items(user story	completion	Estimated hrs	day1	day2	day3	day4	day5	day6	day7	day8	day9	day10	day11
Input imaging	27/12/2021	5	5		0	0	0	0	0	0	0	0	0
Preprocessing image	28/12/2021	5	0	5	0	0	0	0	0	0	0	0	0
Data collection and preprocessing	15/01/2022	5	0	0	2	3	0	0	0	0	0	0	0
Comparison and algorithm implementation	23/01/2022	10	0	0	0	0	5	5	0	0	0	0	0
Edge detection	26/01/2022	0	0	0	0	0	0	0	5	5	0	0	0
Replacing pixel Data	5/02/2022	0	0	0	0	0	0	0	0	0	5	5	0
Writing image Show/save	12/02/2022	0	0	0	0	0	0	0	0	0	0	3	2
total		50	5	5	2	3	5	5	5	5	5	8	2